

---

# ECE540: Homework #4

DAVID KIRBY

DUE: 08 NOVEMBER 2020

---

## Chapter 8 (7<sup>th</sup> Edition)

### Review Questions

#### Section 8.1

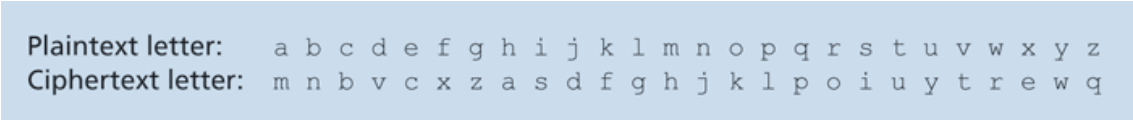
1. R1. What are the differences between message confidentiality and message integrity? Can you have confidentiality without integrity? Can you have integrity without confidentiality? Justify your answer.

#### Section 8.2

2. R3. From a service perspective, what is an important difference between a symmetric-key system and a public-key system?
3. R4. Suppose that an intruder has an encrypted message as well as the decrypted version of that message. Can the intruder mount a ciphertext-only attack, a known-plaintext attack, or a chosen-plaintext attack?
4. R5. Consider an 8-block cipher. How many possible input blocks does this cipher have? How many possible mappings are there? If we view each mapping as a key, then how many possible keys does this cipher have?
5. R7. Suppose  $n = 10,000$ ,  $a = 10,023$ ,  $b = 10,004$ . Use an identity of modular arithmetic to calculate in your head  $(a \cdot b) \bmod n$ .

### Problems

1. P1. Using the monoalphabetic cipher in Figure 8.3, encode the message “This is an easy problem.” Decode the message “rmij’u uamu xyj.”



Plaintext letter:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext letter:	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

Figure 8.3: A monoalphabetic cipher

2. P2. Show that Trudy’s known-plaintext attack, in which she knows the (ciphertext, plaintext) translation pairs for seven letters, reduces the number of possible substitutions to be checked in the example in Section 8.2.1 by approximately  $10^9$ .

3. P9. In this problem, we explore the Diffie-Hellman (DH) public-key encryption algorithm, which allows two entities to agree on a shared key. The DH algorithm makes use of a large prime number  $p$  and another large number  $g$  less than  $p$ . Both  $p$  and  $g$  are made public (so that an attacker would know them). In DH, Alice and Bob each independently choose secret keys,  $S_A$  and  $S_B$ , respectively. Alice then computes her public key,  $T_A$ , by raising  $g$  to  $S_A$  and then taking  $\text{mod } p$ . Bob similarly computes his own public key  $T_B$  by raising  $g$  to  $S_B$  and then taking  $\text{mod } p$ . Alice and Bob then exchange their public keys over the Internet. Alice then calculates the shared secret key  $S$  by raising  $T_B$  to  $S_A$  and then taking  $\text{mod } p$ . Similarly, Bob calculates the shared key  $S'$  by raising  $T_A$  to  $S_B$  and then taking  $\text{mod } p$ .
- Prove that, in general, Alice and Bob obtain the same symmetric key, that is, prove  $S = S'$
  - With  $p = 11$  and  $g = 2$ , suppose Alice and Bob choose private keys  $S_A = 5$  and  $S_B = 12$ , respectively. Calculate Alice's and Bob's public keys,  $T_A$  and  $T_B$ . Show all work.
  - Following up on part (b), now calculate  $S$  as the shared symmetric key. Show all work.
  - Provide a timing diagram that shows how Diffie-Hellman can be attacked by a man-in-the-middle. The timing diagram should have three vertical lines, one for Alice, one for Bob, and one for the attacker Trudy.
4. P12. Suppose Alice and Bob share two secret keys: an authentication key  $S_1$  and a symmetric encryption key  $S_2$ . Augment Figure 8.9 so that both integrity and confidentiality are provided.

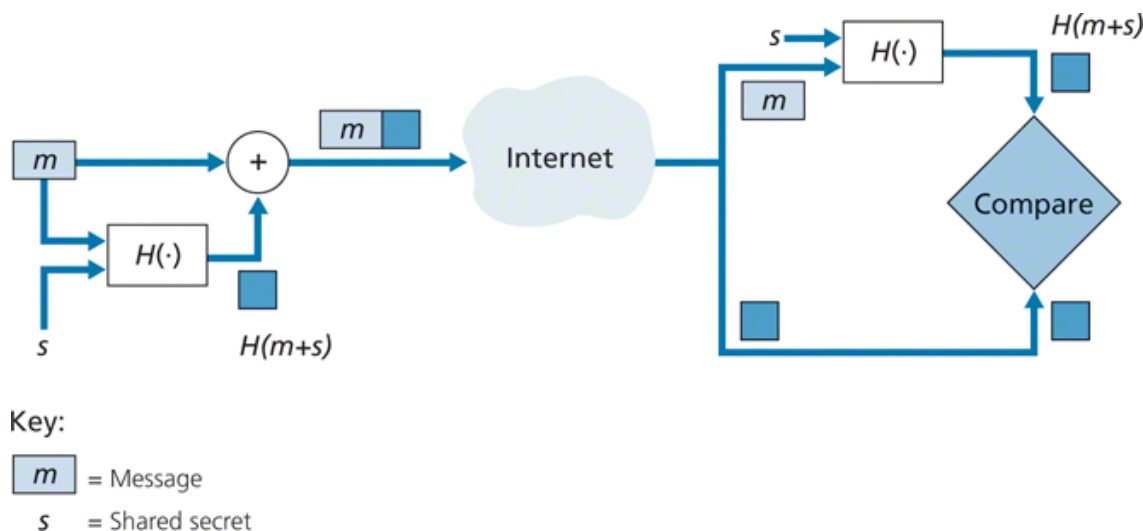


Figure 8.9: Message authentication code (MAC)

5. P15. Consider our authentication protocol in Figure 8.18 in which Alice authenticates herself to Bob, which we saw works well (i.e., we found no flaws in it). Now suppose that while Alice is authenticating herself to Bob, Bob must authenticate himself to Alice. Give a scenario by which Trudy, pretending to be Alice, can now authenticate herself to Bob as Alice. (*Hint: Consider that the sequence of operations of the protocol, one with Trudy initiating and one with Bob initiating, can be arbitrarily interleaved. Pay particular attention to the fact that both Bob and Alice will use a nonce, and that if care is not taken, the same nonce can be used maliciously.*)

6. P19. Consider the Wireshark output below for a portion of an SSL session.
- Is Wireshark packet 112 sent by the client or server?
  - What is the server's IP address and port number?
  - Assuming no loss and no retransmissions, what will be the sequence number of the next TCP segment sent by the client?
  - How many SSL records does Wireshark packet 112 contain?
  - Does packet 112 contain a Master Secret or an Encrypted Master Secret or neither?
  - Assuming that the handshake type field is 1 byte and each length field is 3 bytes, what are the values of the first and last bytes of the Master Secret (or Encrypted Master Secret)?
  - The client encrypted handshake message takes into account how many SSL records?
  - The server encrypted handshake message takes into account how many SSL records?

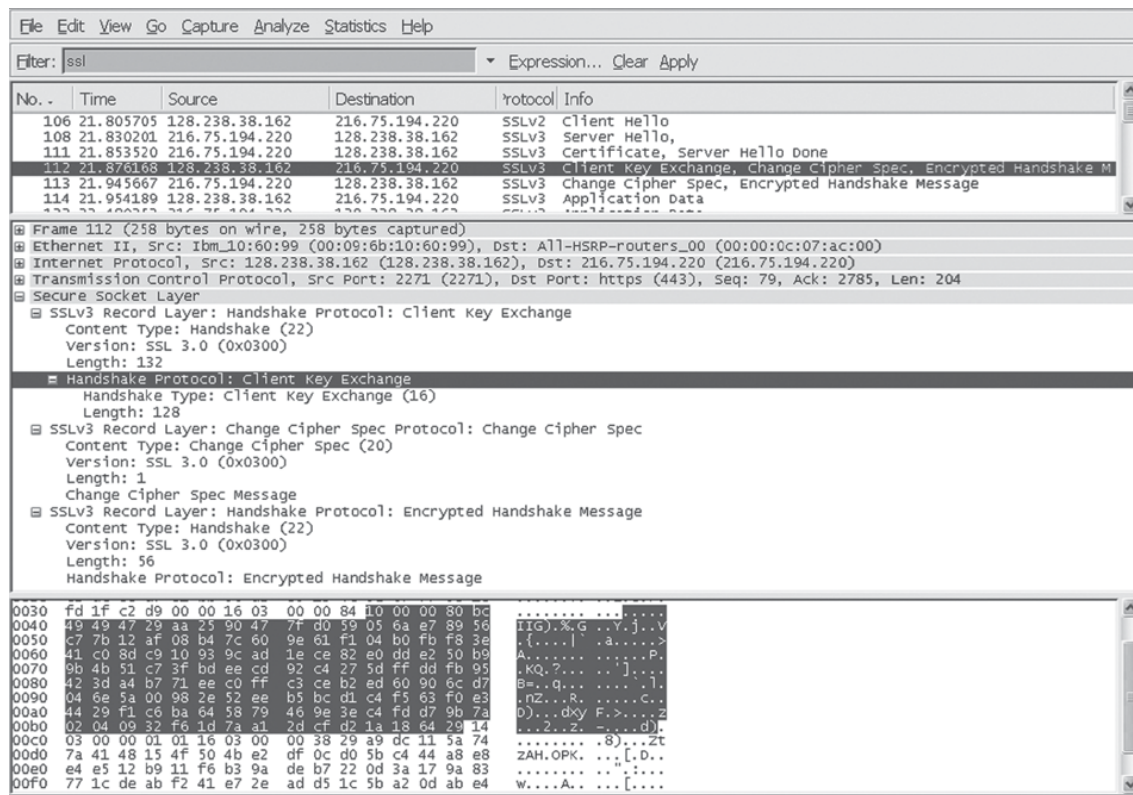


Figure 8.0: (Wireshark screenshot reprinted by permission of the Wireshark Foundation.)

7. P20. In Section 8.6.1, it is shown that without sequence numbers, Trudy (a woman-in-the middle) can wreak havoc in a TLS session by interchanging TCP segments. Can Trudy do something similar by deleting a TCP segment? What does she need to do to succeed at the deletion attack? What effect will it have?